

10/510315
DT04 Rec'd PCT/PTO 01 OCT 2004

**WEB SERVER COMPRISING INTEGRATED AUTOMATION FUNCTIONALITY AND
WITH DIRECT ACCESS TO THE REAL-TIME COMMUNICATION LEVEL OF THE
REAL-TIME ETHERNET**

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application is the US National Stage of International Application No. PCT/DE03/00967, filed March 24, 2003 and claims the benefit thereof. The International Application claims the benefits of German application No. 10214501.6 filed April 2, 2002, both of the applications are incorporated by reference herein in their entirety.

FIELD OF INVENTION

[0002] The invention relates to a web server comprising software modules which are integrated into the web server and an automation system or, as the case may be, a computer program product comprising a web server of said type.

BACKGROUND OF INVENTION

[0003] Servers that are linked via the internet to clients and make information, typically internet pages, available to these clients are called web servers. A web server of said kind is an application which runs on a single computer or in a distributed manner on a plurality of computers. The web server provides centralized storage for data which can be used by many different clients, regardless of the particular location of a client. The term "web server" is used to describe both the software application executing on a computer and also the computer itself. Web servers are nowadays used as a general-purpose means of delivering information in the internet, but also in local area networks which are based on internet technologies. In this arrangement use is often made of the possibility of extending the web servers with expansion modules, for example in order to enable access to databases, forms, etc.

The communication between client and web server usually takes place in accordance with the HTTP protocol (HTTP= Hyper Text Transfer Protocol).

[0004] US 6 061 603 A describes a control system which allows a user to access a programmable logic controller over a communication network such as, for example, the Internet using a web browser. The system includes an interface between the network and the programmable logic controller. This web interface, as it is called, provides internet pages from an Ethernet interface of the programmable logic controller and includes an HTTP protocol interpreter and a TCP/IP stack (TCP/IP = Transmission Control Protocol/Internet Protocol). The web interface thus offers a remote user access to the programmable logic controller via the internet.

SUMMARY OF INVENTION

[0005] The object of the invention is to present a possible solution for implementing an automation functionality which supports different types of communication.

[0006] This object is achieved by means of a web server comprising software modules in which at least a first software module has first means for implementing an automation functionality and second means for directly accessing the real-time communication level of a real-time Ethernet.

[0007] The invention is based on the knowledge that at the present time the conventional automation world has few points of contact with the internet due to the fact that in automation technology use is made predominantly of proprietary protocols. This notwithstanding, the development of web technology is advancing apace without issues relating to automation technology being taken into account. Previous

approaches to integrating dedicated web server functionality into the automation components are in turn based on proprietary solutions for the individual components. Moreover, the performance of solutions of said kind revealed itself to be very limited. The web server according to the invention combines web technology with automation technology in an unexpected manner, in that an expansion module, typically a software module, which is integrated directly in the web server provides the required automation functionality. A web server whose functionality is extended in such a way is capable of processing not only complex tasks of conventional automation technology but also applications on a smaller scale, for example in the consumer domain. The direct integration of the automation functionality into the web server means that existing web implementations can also be used in the processing of the automation function. The second means of the first software module for directly accessing the real-time communication level of the real-time Ethernet also open up a communication path that is independent of web technologies.

[0008] As a result of the connection of the web server according to the invention to a communication network, in particular the internet, on the one hand internet technologies are made accessible to automation technology and on the other hand an end-to-end connection of the automation components to the communication network or, as the case may be, the internet is achieved.

[0009] The use of internet protocols for communication between the software modules themselves and for communication between the software modules and components located outside of the web server enables the standardization of the components of the web server with regard to their communication interfaces. Expansion of the web server with further modules is facilitated, since said

modules can be used without complex and time-intensive adaptation to proprietary protocols. Example of typical internet protocols are HTTP and FTP (File Transfer Protocol).

[0010] In an advantageous embodiment of the invention the web server is provided for the configuration and administration of the software modules.

[0011] The web server according to the invention is easily scalable and so can be used, according to a further advantageous embodiment, for controlling the components of an industrial automation system if the first software module has a connection to the industrial automation system.

[0012] In order to use available internet security mechanisms it is proposed that the web server has a connection to the internet via a firewall. In the case of the already typical web server expansions integrated into an automation component, the security mechanisms required in the internet usually cannot be implemented because of the limited scope for modification.

[0013] The connection of the web server to a communication network, in particular the internet, can be advantageously used to support the automation functionality if the web server is linked via a communication network to a web browser as the control and monitoring system for the automation system controlled by the first software module. This control and monitoring system can also be used for project planning, for configuration, for programming, and for rolling out software updates, in other words generally for data communication and data representation.

[0014] In order to enable web technologies to be used also for real-time applications it is proposed that the web

server has a real-time operating system. In particular when deployed in process automation, the automation components used must have real-time capability. This requirement can be met by coupling of the first software module to the real-time operating system. The real-time operating system can be used in addition to a non-real-time-capable operating system (part) or as a standalone operating system.

[0015] The invention will be described and explained in more detail below with reference to the exemplary embodiments represented in the figures, in which:

BRIEF DESCRIPTION OF THE DRAWINGS

- FIG 1 shows a system comprising web servers which are interconnected via the internet and fulfill different tasks,
- FIG 2 shows a web server comprising automation functionality,
- FIG 3 shows a schematic view of the structure of a web server comprising automation functionality,
- FIG 4 shows a system comprising web servers which are interconnected via the internet and a proprietary connection, and
- FIG 5 shows a system comprising the combined use of a web server with automation functionality, an automation device, and a conventional web server without automation functionality.

DETAILED DESCRIPTION OF INVENTION

[0016] FIG 1 shows different web servers 3, 10, 15, 20, 24 which are connected to one another directly or indirectly via the internet 1. A first web server 3 communicates directly with the internet 1 via a connection 2. The first web server 3 includes an expansion module 4 which is connected via a connection 5 to an input/output module 6 of an automation system. A second and a third web server 10, 15

are connected to the internet 1 via connections 9, 14, a firewall 8 and a connection 7. The second web server 10 has an expansion module 11 which has a connection 12 to a converter 13. The third web server 15 includes an expansion module 16 which controls a drive 18 via a connection 17. The reference numeral 20 identifies a fourth web server, referred to as an embedded web server, which is connected directly to the internet 1 via a connection 19 and has an expansion module 21 which controls a valve 22. The fifth web server 24 shown in FIG 1 possesses no automation functionality whatsoever and communicates with the internet via a connection 23. A web browser 26 is connected to the internet 1 via a connection 25.

[0017] The idea on which the invention is based will be explained below with reference to FIG 1. A web server is a process running on a computer - or also distributed over a plurality of computers - and typically supplies very many clients (web browsers on different devices) with information. This information can either reside statically on the web server or else be generated dynamically by further utility routines. Typical communication partners connected via the internet 1 are therefore web servers in the embodiment of the fifth web server 24 and web browser 26. The fifth web server 24 provides information, generally internet pages, via the internet 1 in response to a request from a web browser 26. The idea of the invention is thus to configure a standard web server of this type through expansion by means of software modules in such a way that it can also solve automation tasks. The first web server 3 includes an expansion module 4 which takes on the functions of a programmable logic controller (PLC). Toward that end, the expansion module 4 as part of the web server 3 is additionally connected to an input/output module of an automation system by means of a connection 5. The first web server 3 therefore serves not only to deliver information

into the internet 1 via the connection 2, but can also, thanks to the integration of the expansion module 4, execute complex control tasks which in the prior art were previously only executable by means of independent programmable logic controllers. A further exemplary embodiment of the web server according to the invention is shown in FIG 1 in the second web server 10 which has an expansion module 11 comprising CNC (CNC = Computer Numerical Control) functionality. Via the expansion module 11, the second web server 10 controls a computer-controlled machine tool 13 (CNC machine tool) which is used for fast and precise production of complex turned and milled parts. Complex controls of said kind are usually executed by computers specified explicitly for that purpose. The control of a drive 18 which is handled by the third web server 15 in the exemplary embodiment reveals itself as a similarly complex control task. For this purpose the third web server 15 includes an expansion module 16 which assumes the demanding tasks of controlling and/or regulating the drive 18. In order not to pay for the advantages of using web technologies by the second and the third web server 10, 15 at the expense of the disadvantage of absence of security, the web servers 10, 15 are linked to the internet 1 via a firewall 8. The firewall 8 effectively prevents illegal accesses via a communication connection 7 to one of the web servers and therefore to the drive 18 or the machine tool 13. In a further exemplary embodiment of the invention shown in FIG 1, the web server comprising automation functionality is what is known as an embedded web server 20 which includes, in the form of expansion module 21, a temperature controller for controlling a valve 22. This embedded web server 20 is implemented for example as a single-chip solution within a personal computer (PC). In addition to the automation functionality of the expansion modules, each of the described web servers 3, 10, 15 and 20 also provides all the functionalities and therefore all the advantages of a

standard web server 24. Thus, the web browser 26 connected via the internet 1 can also make use of web technologies to access the web servers 3, 10, 15 and 20 that have been extended with automation functionality and so be used, for example, as a control and monitoring system. The exemplary embodiments illustrated in FIG 1 clearly show the better scalability of the solution proposed here in contradistinction to traditional approaches. The web server can be embodied as a single-chip web server with hardware add-on (e.g. in the consumer domain) right through to a high-performance web server featuring SoftPLC and Office software.

[0018] FIG 2 shows an exemplary embodiment of a web server comprising automation functionality in a schematic view. A web server 33 is connected via a connection 32 and a TCP/IP stack 31 to a TCP/IP connection 30. The web server 33 includes a first expansion module 34 which is embodied as a database module and has access by means of a connection 35 to a SQL7 server 36 (SQL = Structured Query Language). A second expansion module 37 possesses automation functionality and communicates via a connection 38 with an industrial process 39. The expansion module 37, a software module, is coupled via an interface (not shown in the figure) to the connection 38 and therefore to the hardware components of the automation system for controlling the industrial process 39. As front-end software, the TCP/IP stack controls the accesses to a network card (not shown in the figure) which has access to the TCP/IP connection 30 and makes the TCP/IP protocol available to the accessing processes.

[0019] FIG 3 shows the schematic view of the structure of a further exemplary embodiment of a web server comprising automation functionality. Software components of the web server are depicted. An operating system 50 of the computer

on which the web server runs includes a standard operating system 51 as well as a real-time operating system 52. A web server kernel 54 is superimposed on top of the underlying operating system 50 using the intermediary of a porting 53. The web server kernel 54 provides standardized interfaces for coupling the software components and forms the basis for various software expansion modules. A first expansion module 55 serves to provide web pages, while a second expansion module 56 serves as an XML parser. A Profibus access 58 is connected by means of a third expansion module 57. A fourth expansion module 59 provides Java functionality, while a fifth expansion module 60 processes signals of a webcam. A sixth expansion module 64 serves for processing XML data. A seventh expansion module 61 comprising automation functionality has a connection 62 to a process and a connection 63 to an interface 65 to the real-time operating system 52. A system is regarded as real-time-capable if it can react to random external events within a specifiable and guaranteeable time. Reaction times in the microsecond range are typical and essential in industrial automation systems. Real-time processes can use all the services which are also available to other processes.

[0020] The operating system 50 has direct access to the resources of the computer, such as, for example, memory and computing time. When a command is sent off or a program invoked, the required program code is loaded into a main memory and started as a process. Processes have no access to the resources, but request these from the operating system in each case.

[0021] The direct integration of the automation functionality into the web server in the form of the seventh expansion module 61 makes the performance, services (e.g. autotopology, SSDP, SNMP, e-mail etc.) and the openness of the internet accessible to the automation world and enables

the additional above-described advantages to be achieved. The seventh expansion module 61 on the one hand implements the automation solution and on the other hand exchanges information via the web server and is configured and administered by said web server. In contrast to this, in the case of a so-called SoftPLC (= software simulation of a programmable logic controller) the automation function is not integrated into the server, but is installed in parallel thereto, possibly connected via a communication interface. Integration in the server means in particular that an expansion module is loaded, configured, started and terminated directly by the web server. An expansion module of this kind is frequently referred to also as an "extension". The web server kernel 54 of the web server according to the invention serves as a common platform for the expansion modules. This facilitates in particular the configuration of the software expansion modules and their reuse in other applications. The expansion modules are not coupled by means of proprietary interfaces or interfaces which have been programmed out, but are connected by means of standardized interfaces, for example API (Application Programming Interface) or CGI (Common Gateway Interface). API is a formally defined interface via which application programs can use system services (network, operating system, etc.) or services of other application programs. CGI describes a standard interface between a web server and programs. As means for implementing an automation functionality, the seventh expansion module 61 has regulation and/or control means for regulating and/or controlling components and processes of an automation system. Said regulation and control means for controlling an automation system are typically embodied as software processes which are executable in the expansion module.

[0022] FIG 4 shows a system with web servers 70, 71 interconnected via the internet 80 and a proprietary

connection 78. A web server 70 comprising the automation module 72 is connected to the internet 80 via a connection 76 and the web server's TCP/IP stack 77. The automation module 72 communicates with an industrial process 74 via a connection 73. The automation module 72 integrated into the web server additionally has direct access by means of the connection 75 to the real-time communication level of the real-time Ethernet. The automation module 72 is therefore extended to include access to the real-time communication level of the real-time Ethernet. This is generally the underlying real-time Ethernet TCP/IP stack in the fourth layer (also referred to as "Layer 4") of the ISO/OSI model described below. A real-time Ethernet connection 78 between the TCP/IP stack 77 of the web server 70 and the TCP/IP stack 79 of a further web server 71 is used for communication by means of a TCP/IP-based real-time Ethernet protocol. The web server 71 also has an automation module 83 which is coupled to the TCP/IP stack 79 via a connection 82.

[0023] FIG 5 shows a system comprising the combined use of a web server 70 with automation functionality, an automation device 93, and a conventional web server 95 without automation functionality. A web server 70 comprising the automation module 72 is connected to the internet 80 via a connection 76 and the web server's TCP/IP stack 77. The automation module 72 integrated into the web server additionally has direct access by means of the connection 75 to the real-time communication level of the real-time Ethernet. A real-time Ethernet connection 78 between the TCP/IP stack 77 of the web server 70 and the TCP/IP stack 91 of an automation device 93 is used for communication by means of a TCP/IP-based real-time Ethernet protocol. The conventional automation device 93 is coupled to its TCP/IP stack 91 via a connection 82. FIG 5 also shows a conventional web server 95 without automation functionality

which is coupled to the internet 80 by means of its TCP/IP stack 94.

[0024] Certain automation tasks require real-time-capable communication (fast, equidistant, deterministic; see above) of a kind which the internet protocols and also standard Ethernet cannot provide. The problem of protocols that from this point of view are not always sufficiently powerful is resolved in that access to the underlying real-time Ethernet protocol stack is enabled for the automation modules 72, 83 of the web server 70, 71. Proprietary, high-speed, real-time communication can take place via this route. This is of course relevant in particular in LAN environments (LAN = Local Area Network). The real-time requirements of the automation module 72, 83 are handled by means of this access. Administrative tasks such as configuration, diagnostics, etc. or also "surfing the web" (meaning normal internet access) are handled via the non-real-time part of the web server 70, 71.

[0025] To provide clarification, the basic principles of the ISO/OSI reference model referred to in the description and of the TCP/IP protocol are explained below. An architecture model developed by the International Standards Organisation (ISO) is frequently used as a basis for describing the structure and function of protocols for data communication. This basic reference model, known as the Open Systems Interconnect (OSI) reference model, creates a frame of reference for dealing with issues arising from the field of data communication. The ISO basic reference model consists of seven layers. Each of these layers defines certain functions of the data communication protocols which are executed during the exchange of data between applications over an intermediate network. Each individual layer does not define a protocol but rather represents a data communication function which can be executed by any number of protocols.

Each layer can contain a plurality of protocols, each of which provides such services as are required for fulfilling the function of said layer. The following model results:

[0026] The bottom, first, layer defines the physical characteristics of the transmission paths and is therefore known as the physical layer. The second layer, known as the data link layer, ensures reliable transmission of the data over the physical connections. The third layer, the network layer, administers the connections between the computers in the network for the higher layers. The fourth layer, called the transport layer, ensures error-free data transmission by means of error detection and correction. The fifth layer, known as the session layer, administers the connections between the applications. The sixth layer, the presentation layer, standardizes the format of the data on the network. Finally, the highest and seventh layer, the application layer, consists of the applications by means of which the network can be used.

[0027] TCP/IP (Transmission Control Protocol/Internet Protocol) is the standard protocol stack for connecting to the internet, not only in the UNIX world but also on PCs (DOS, Windows, etc.). Advantages of TCP/IP include the open, vendor-independent protocol specifications, the independence from a specific network medium, and the standardized interface to application programs. Compared with the ISO/OSI model, the TCP/IP protocol architecture is based on four layers. The bottom layer is known as the network access or network interface layer and contains routines for accessing physical networks. It corresponds to the bottom two layers of the ISO/OSI model. The second TCP/IP layer, the internet layer, defines the structure of datagrams and routes data and therefore corresponds to the third OSI layer. The third TCP/IP layer, the (host-to-host) transport layer, provides end-to-end data services and corresponds to the fourth layer

of the ISO/OSI model. The fourth TCP/IP layer, known as the application layer, contains applications and processes which access the network and so corresponds to the top three layers of the ISO/OSI model. The data travels down through the individual layers and at each lower layer has added to it a new header containing control information (this process is known as capsulation). When the data is passed through the layers from bottom to top, this additional information is stripped out again.

[0028] The proposed web server is part of a system of distributed applications that is constructed on the basis of a client-server architecture. In a system of this kind it is the task of a server as the provider of a service to perform calculations or other internal processes in response to requests from a client and to formulate its results as protocol-compliant answers and pass them on to the requesting client. The term "client" in this case refers to a device or a process which makes use of the service of one or more servers. Usually, therefore, the server makes a service available passively and waits for a client to submit a request to it. The client, on the other hand, makes no services available but instead avails itself of services provided by a server. A server as the provider of a service can reside on the same device as the client or on a different device which is accessible via a network (for example the internet). The client-server communication obeys certain rules and formal descriptions, called protocols. It is an essential requirement for successful communication between client and server that both sides use the same protocol. Such a protocol typically specifies the communication channels and the formats used for login, information exchange, request, response and logoff. Not all of these steps have to be specified explicitly in every case if they are not relevant to the purpose of the application. Protocols are specified at the most diverse abstraction

levels and normally build one on top of the other. In that case the structure is referred to as a layer model (e.g. the ISO/OSI layer model described above). Whereas the lowest layers regulate the communication of hardware and devices - electrical signals, cables or radio frequencies and their characteristics are specified - the middle layers are concerned with the building of network topologies (address structures and their resolution, routing, and error correction). In this case the network layer (e.g. IP = Internet Protocol) is often separated from the transport layer (e.g. TCP = Transmission Control Protocol). The topmost layers are referred to as the application layer. At this level it is specified how actual client-server applications communicate with one another. Examples of such application layer protocols are HTTP (Hyper Text Transfer Protocol), FTP (File Transfer Protocol) and SMTP (Simple Mail Transfer Protocol).

[0029] In summary, the invention therefore relates to a web server which comprises software modules 72 that are integrated into the web server 70 and which supports different types of communication. At least a first software module 72 has first means for implementing an automation functionality and second means for directly accessing the real-time communication level of a real-time Ethernet.